



Available online at <http://scik.org>

J. Math. Comput. Sci. 11 (2021), No. 1, 61-73

<https://doi.org/10.28919/jmcs/5094>

ISSN: 1927-5307

SPARSE REPRESENTATION USING COMPRESSED SENSING VIA DEEP LEARNING

D.M. ANNIE BRIGHTY CHRISTLIN^{1,*}, M. SAFISH MARY²

¹Department of Computer Science, Manonmaniam Sundaranar University, Tamilnadu, India

²Department of Computer Science, St Xaviers College, Tamilnadu, India

Copyright © 2021 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: Since the patients are not static during the MRI acquisition, the image formation process can create some artifacts that could reduce the photograph quality. The Compressed Sensing (CS) mechanism is hired for reconstructing the unique picture from the given sparse data. Accordingly, CS can be applied to reduce the acceleration time for an MRI test considering the patient's health. So the sensing process is carried out by way of a projection matrix for reconstructing the sparse signals from a few numbers of measurements. The CS guarantees the healing of an authentic picture with an excessive possibility based totally on random gaussian projection matrices. However, sparse ternaries projections (Latin word)-1, zero, +1 are more apt for hardware implementation. The proposed deep learning technique is hired in this article to acquire very CS sparse ternary projections. The STDAENN architecture incorporates the sensing layer for the projection matrix and a reconstruction layer for non-linear sparse matrix continuously by auto-encoder. The overall performance of the proposed STDAENN method is compared with the present strategies primarily based on the imply top signal-to-noise ratio (PSNR) to check the picture nicely.

Keywords: compressed sensing; deep learning; sparse ternarius (Latin word) projections.

2010 AMS Subject Classification: 65Y04.

*Corresponding author

E-mail address: anibrighty@gmail.com

Received October 8, 2020

1. INTRODUCTION

CS merges the process of compression and sensing or acquisition. One of the properties such that sparsity is used to regain the signal has sampled at an appreciable decrease rate than the necessities of the Shanon/Nyquist theorem [1]. In the scientific regions such as MRI and CT, audio or video or picture [2], the effective processing and analysis of excessive-dimensional facts the usage of CS is very important. Consider the compressed signal $x \in \mathbb{R}^n$ is held by CS sensing mechanism. It is examined with the help of projection matrix. So the linear measurements are given below:

$$l_1 = l_2 = \dots l_m = l = \Phi x \quad (1)$$

Where $\Phi \in \mathbb{R}^{m \times n}$, Here $m \times n$ is the projection matrix and $l \in \mathbb{R}^m$ is the vector. In Sparse Sampling, anticipate that either x is a sparse signal or that x has a sparse representation for a suitable basis rely the performance coefficients of the received signal for getting the under-determined linear system

$$l = \Phi \psi u \quad (2)$$

The l is obtained through the sparse vector. This is referred to as the solution of the l_1 minimization problem.

$$\begin{aligned} \min \|u\|_1 \text{ subject to } l = \Phi \psi u \\ u \in \mathbb{R}^n \end{aligned} \quad (3)$$

The predictable CS method basis pursuit [3] provides the time complexity of the n measurements $o(s \log(n/s))$ within the n -dimensional-sparse signal based on random Gaussian matrices[4]. One important trouble that random matrices are normally considered to construct hardware is very tough. Additionally, the arbitrary matrices are treated with sign vectors of high dimension. If there is no rapid matrix multiplication algorithm, Deep Learning knowledge may be used to examine the couple of levels of statistics representation in image processing effectively. The known Deep Learning method has been used for image tremendous-resolution [5], CS [7][8]and image denoising [6].

In this article, the deep learning method is utilized to find out an optimized projection matrix and non-linear reconstruction plotting from obtained measurements to the original signal

for implementing the hardware by an apt projection. The proposed network architecture are implementing the hardware through the sparsity and binary constraints on required by sparse $s \in \{0, -1, +1\}$ to be focused. The image patches are tested by the proposed STDAENN network and the images are acquired in a block-based manner using the learned projection matrix. The result of the STDAENN method shows that the image with high quality in the reconstruction can be achieved with 0.05 projections nonzero $\{-1, +1\}$ entries experimentally.

The review of Compressed Sensing is discussed in Sec 2. The Compressive Sensing via Deep learning is highlighted in Sec 3. The experimental results are analyzed in Section 4 for describing the improved CS approach in image reconstruction. Hence The STDAENN method is focused to improve the image quality.

2. LITERATURE REVIEW

2.1 Compressed Sensing via Deep Learning

The healing for regaining the authentic sign from compressed sensing length is carried out with the aid of deep learning [7][8]. The Stacked Denoising Autoencoder (SDA) is used for plotting a map of non-linear sensing operator and reconstruction [7]. The block-based CS completely linear sensing and non-linear reconstruction method are carried out using this absolutely-related community via Deep Learning. In the training time, the sensing matrix and the reconstruction methods are optimized to offer the final result of the proposed method primarily based at the wonderful reconstruction and evaluation time [8]. Here, the projections are dealt with a dense matrix [7] [8]. So the projection matrix handled as a sparse matrix element with the range of -1, 0, +1. So it offers the fast computation at some stage in acquisition and also improves the hardware implementation.

While a DNN is mapped with binary weights -1, +1, this binary connect technique is used in the time of the forward and backward propagations. So the collected gradients of the stored weights are regained [13].

At runtime, the formation process reduces the memory length. However, a few scaling

elements are introduced for compensating the lack of introducing the burden binarization using BNN [14] and additionally this technique is used to compress the pre-retained community [15]. The proposed approach [16] is used for analyzing the binary weights, links and also bringing a carefully related community. This mechanism is elevated [17] for compressing DNN by connection pruning, Huffman coding, and weight quantization.

The proposed sparse technique is used for imposing the sensing layer by the binarization method that yields a fairly sparse ternarius (Latin word) projection matrix. So the learned projections allowed speedy computations during acquisition. So this method is apt for hardware implementation. The primary layer is constructed for sensing the signal in this approaching network, which corresponds to the linear projection matrix and allows the reconstruction module to be the non-linear for obtaining an excessive overall performance.

3. CS VIA DEEP LEARNING

Now an efficient technique of CS is proposed for the clinical picture particularly STDAENN (Stacked Denoising Autoencoder Neural Network), applied to map the sensing and reconstruction layer for the non-linear measurements for acquiring an excessive overall performance. This phase describes the STDAENN architecture, observed by employing the stairs for the education set of rules

3.1 STDAENN Architecture

The STDAENN architecture is hired in fig. 1. It carries a sensing and reconstruction layer. The sensing layer (SeLa linear layer) presents in the left-half of the STDAENN architecture and the reconstruction layer (non-linear layer) presents in the right-hand side. The community takes an input as the vectorized picture patches (IM) of length $n = S^2$. The sensing layer sends the n -dimensional input signal x to the m -dimensional domain.

Thus, the variety of gadgets in this layer is $m = s^2v$, in which the sensing value v presents between 0 and 1. The weights of sensing layer $\Theta_{abx} \in -1, 0, +1$ $m \times n$, which corresponds to the projection matrix $\Theta_{abx} = \Phi t$.

In the reconstruction module, the scaling layer has the input as the output of the sensing layer by way of the found out factors α linearly. This residue consists of “m” hidden units that are linked “one-to-one” for “m” hidden units of the sensing layer and it's far used to make amends for the loss which changed into made by way of binarizing the sensing weights. The projection matrix is implemented inside the Sensing Layer and the learned scaling elements are stored in the reconstruction module. The Scaling Layer (ScLa) is followed by way of m Hidden Layers (HLa). Those hidden layers are the use of the rectified linear unit (Relu) activation feature [17]. The output layer (OLa) is a linear completely related layer which has the size is equivalent to the input size. The HLa's within the reconstruction module is fully connected and observed by a batch-normalization layer [18].

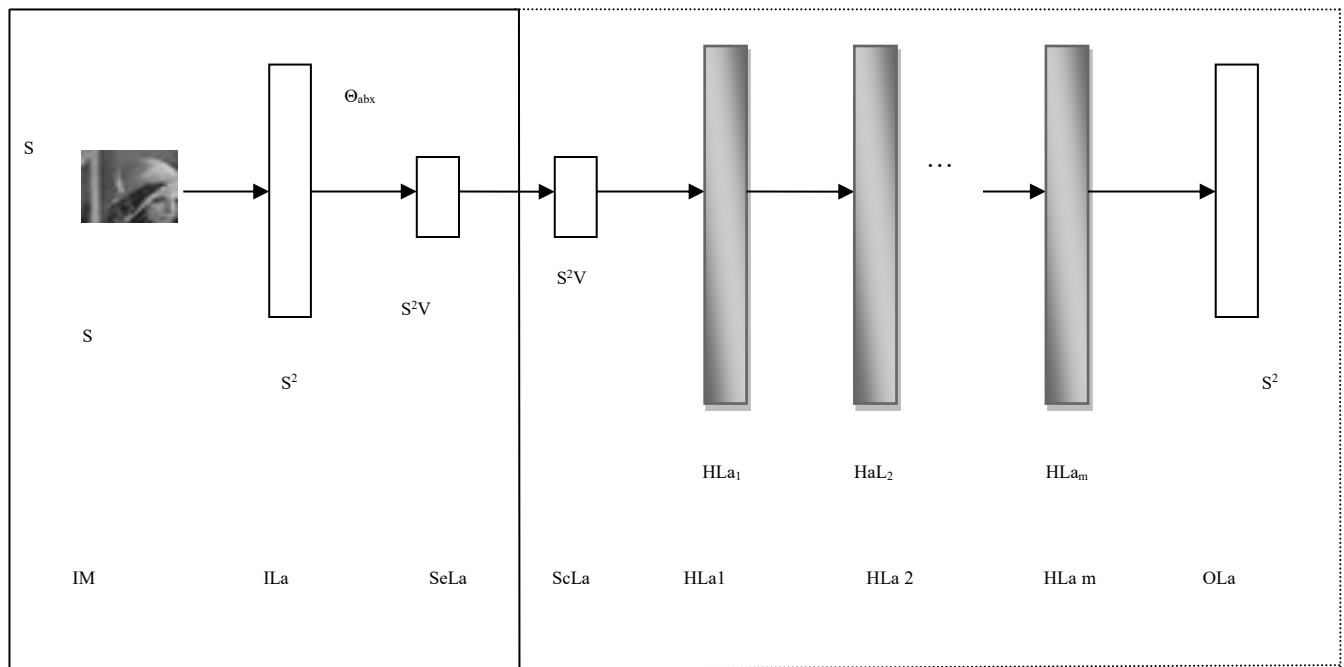


FIGURE1. STDAENN Architecture

3.2 STDAENN Process

In general, network training uses the standard mini-batch gradient descent method. z_i, z^{\wedge}_i denotes the input and reconstructed patches, respectively, with $z_i, z^{\wedge}_i \in \mathbb{R}^n, n = S^2$. The MSE is obtained as the loss function for the difference between the input and the reconstruction

$$L = \frac{1}{N} \sum_{i=1}^N \|z_i - z^{\wedge i}\|_2^2 \quad (4)$$

Here N is the no. of pattern patches.

Algorithm:

1. A sparsifying and binarization steps are delivered on the sensing layer.
2. The continuous-valued sensing weights $\Theta \in \mathbb{R}^{m \times n}$ are sparsified to get the sparse weights $\Theta_{ax} \in \mathbb{R}^{m \times n}$.
3. For this step, two among the sparse weight Θ_{ax} retain only the entries that correspond to the top- K biggest absolute values as Θ and set all the remaining entries to zero. This technique is termed as a K -Select function. The selection of the pinnacle weights can be assigned column-wise, row-wise or over the whole matrix. However, because the scaling layer replaces column-wise operator on the sensing weights, the selection from the top K entries can be made in a column-wise manner.
4. In the sensing layer, every neuron are connected to $K = S^2\gamma$ factors as the input signal, where γ is the sparsity ratio.
5. A sparse binary mask $M_a \in \{0,1\}^{m \times n}$ is built with entries equal to 1 that corresponds to the largest weights in Θ implementation wise.
6. The sparse sensing weights are updated in accordance with $\Theta_{ax} = M_a \Theta_a \Theta$ the place Θ_a represents the Hadamard product.
7. The sparse continuous-valued weights $\Theta_{ax} \in \mathbb{R}^{n \times m}$ are mapped to the sparse binary weights $\Theta_{abx} \in \{-1,0,+1\}^{m \times n}$. This process is required for binarization. However, both the sparse and binarization step creates some loss that can be recovered in the process of reconstruction.
8. The sparse binarized weights Θ_{abx} are mapped inversely to the continuous weights Θ using the layer weights $\alpha_x \in \mathbb{R}^m$. Thus, the output of the scaling layer $\alpha_x \Theta_{abx}^T y$ is an approximation of $\Theta^T y$, which corresponds to the non-stop projections Θ^T .

Mathematical Representation:

Let $\theta(j)$ and $\theta_{abx}(j)$ are the j th columns of Θ and Θ_{abx} , respectively. $\theta(j)$ corresponds to the dense non-stop weights of the j th hidden units in the sensing layer. Let approximate $\theta(j)$ with $\theta_{abx}(j)$, the place $\alpha x \in V^+$ is a scale factor, corresponding to the j th entry of the scaling weights αx . The values of $\theta_{abx}(j)$ and $\theta(j)$ can be determined by minimizing the following imply rectangular error for $\theta_{abx}(j), \alpha x(j)$

$$E = \|\theta(j) - \alpha x(j) \theta_{abx}(j)\|^2 \quad (5)$$

By expanding (5), we have:

$$E = \theta(j)^T \theta(j) - 2 \alpha x(j) \theta(j)^T \theta_{abx}(j) + \alpha x(j)^2 \theta_{abx}(j)^T \theta_{abx}(j) \quad (6)$$

Where $\theta(j)^T$ and $\theta(j)$ are constant. As $\alpha x(j)$ is taken as the sparsity constraint [15], the optimal sparse binary vector $\theta_{abx}(j)$ is obtained as a solution to the problem is given below:

$$\theta_{abx}(j)^* = \text{argmax}(\theta(j)^T \theta_{abx}(j)) \theta_{abx}(j)$$

$$\text{such that } \theta_{abx}(j) \in \{-1, 0, +1\}^{m \times n}$$

$$\text{supp}(\theta_{abx}(j)) = \text{supp}(\theta_{ax}(j)) \quad (7)$$

where $\theta_{ax}(j)$ is the j^{th} column of Θ_{sx} , and $\text{supp} \Theta$ denotes the nonzero entries position in the treated vector. The result of (7) is a vector, which contains the sign of $\theta_{abx}(j)$. After obtaining the optimal $\theta_{abx}(j)$, the optimal $\theta(j)$ is obtained by making the derivative of eqn equal to zero.

Consider $\theta_{abx}(j)^T \theta_{abx}(j) = K$,

the optimal $\alpha(j)$ is given by:

$$\alpha(j) = 1/K (\theta(j)^T \theta_{abx}(j)) = 1/K (\sum_{i=1}^n \theta(j)_i \theta_{abx}(j)_i) = 1/K (\|\theta_{ax}(j)\|_1) \quad (8)$$

After the completion of sparsifying and binarization steps, the resultant Θ_{abx} is sparse, which has the K nonzero entries $\{-1, +1\}$ in each of its columns. During the forward and backward propagation, the sparse binary weights Θ_{abx} are used. This process is done after this algorithm works with binary weights [13,14,15]. An excessive precision weights Θ are used for making the small modifications of the weights after every update step through the replacement of the parameter W_t . In the proposed training, Θ is updated by the gradient of loss characteristic for Θ_{abx} . This replacement of parameter is referred to as an Adam parameter update. Θ_{abx} contains only discrete

weights such as $\{-1,+1\}$ during the gradient of loss which is concerned still lies in the continuous domain.

Finally, the loss, newly updated weights of continuous and reconstruction domain such as Θ^t , W^t the sparse binary sensing weight Θ_{abx}^t and the scaling layer weight αx^t are obtained.

The overall performance of the proposed approach is in contrast with the existing techniques primarily based on the PSNR using a sparse ternary projection to take a look at the image quality.

4. EXPERIMENTAL RESULT ANALYSIS

The ILSVRC2012 set [19] contains 2000 samples of 50 K images are taken for training this network. The proposed model is examined on the two checking out set of 256X256 resolution images. The first testing set consists of 10 images, taken from the ILSVCR2014 [19] dataset, and is given by using the authors of [7]. The second testing set is composed of 50 images chosen from the LabelMe dataset [20] randomizely. All the images had been converted to gray scale in this experiment. These are run in 32 x 32 dimension pixels of small image patches for reducing the computational overhead. The 2000 patches are sampled randomly to construct the coaching set. In the training time, the input patches have been preprocessed through the well known deviation. This network is skilled the use of the proposed training algorithm with the Adam parameter replace [21] a batch size of 5000, 50 epochs and 0.01 decomposing by means of an aspect of 0.6 each and every 5 epochs. The training samples have been randomly shuffled after each epoch. The l_2 regularization is applied on the reconstruction module, with a weight as 0.001 for keeping away from out-of-fitting. During the testing phase, these overlapping patches are sampled from each test image with a stride of 2 pixels and decided the ultimate image reconstruction as the average of the patches' reconstruction. These strategies are evaluated by PSNR value which is expressed in dB. The variety of non-linear hidden layers are assigned empirically L to 2, each with 2048 hidden devices are referred to this community architecture. So this configuration produces a good outcome between training time and reconstruction quality.

Evaluation of PSNR based on sensing rate

This network is trained and examined with different sensing rates using $\gamma = 0.1$. The small changes are appeared R in $[0.1,0.3]$. The mean PSNR values on the first testing set are proven in Table-1. So the resultant better reconstruction fine is received with large sensing quotes because greater data from the sign is retained in the measurements.

Table-1. Reconstruction Performance based on varying Sensing rate R ($\gamma=0.1$)

R	0.1	0.15	0.2	0.25	0.3
PSNR	30.85	30.83	31.0	31.28	31.1

Evaluation of PSNR based on Sparsity ratio

The community is trained and tested with unique sparsity ratios, the usage of $R = 0.25$ and exceptional γ . The suggest PSNR values on the first testing set are presented in Table-2. The dimension of input is 32×32 and, for $R = 0.25$, the end result is bought as $\Theta_{abx} \in \{-1,0,+1\} 1024 \times 256$. With $\gamma \in \{0.001,0.005,0.010,0.050,0.100,0.300,1.00\}$, the number of nonzero entries in every column of Θ_{abx} (i. e. K) is 1, 5, 10, 51, 102, 307,1024 respectively. An acceptable reconstruction overall performance can be accomplished using extraordinarily sparse projection matrices with only 0.1% nonzero entries ($\gamma = 0.001$). The value of γ altering from 0.001 to 0.05 significantly improves the performance. The network reaches its peak performance with $R = 0.25$ and performs slightly worst with $\gamma \in \{0.10,0.30\}$. It should be pointed out that $\gamma \in \{0.001,0.005\}$ there are 256 and 1280 nonzero entries in the projection matrix, respectively. The former is no longer enough to fully deal with the 1024-dimensional input signal. Hence, this is the motive for an apparent overall performance start when growing γ to 0.005. During training, the community experiences over-fitting with $\gamma \in \{0.1,0.3\}$. So $\gamma = 0.05$ offers better performance than $\gamma \in \{0.1,0.3\}$. As a result, the proposed sparse binary constraints can be viewed as a greater regularizer to the network.

Table-2. Reconstruction Performance baed on varying Sparsity ratio γ (Choose R=0.25)

γ	0.001	0.005	0.01	0.05	0.1	0.3
PSNR	30.88	31.02	31.16	31.38	31.28	30.96

Since the proposed algorithm executes CS via deep learning, the next experiment compares with the method of [7], which uses a stacked denoising autoencoder to jointly learn the sensing layer and the reconstruction. The Overlapping Non-Linear Stacked Denoising Autoencoder(O-NL-SDA)[7] is treated for comparing with the proposed algorithm. This algorithm uses a non-linear sensing mechanism, with overlapping image patches of 32 X 32size. The result of O-NL-SDA on the first testing set[7] is taken. To obtain the results on the second testing set, an O-NL-SDA model which trained with the training set using the proposed configuration [7]. The results are obtained with a conventional reconstruction algorithm, namely, Basis Pursuit (BP) [3], using random ternary projections are also presented. Since the constraints imposed on the matrix dimensions, the sparse binary and ternary constructions proposed in [22, 23]. In $R=0.25$ the value of PSNR is better than all. Choose $\gamma= 0.01$ for the proposed method since it yields a better performance while producing a highly sparse projection matrix. The comparison between the selected methods on the first testing set is shown in Table-3. On this testing set, the mean PSNR values for O-NL-SDA [7], BP [3] and STDAENN are 26.34, 22.07 and 27.24 dB, respectively. The STDAENN algorithm gives better results than BP and O-NL-SDA. Despite of having a sparse ternarius(Latin word) matrix of only 0.01 of non-zero entries, the proposed method outperforms O-NL-SDA in terms of the recovery performance. The reconstruction is implemented using a feed-forward neural network that can perform orders of magnitude faster than a convex optimization solver relative to the speed of the reconstruction [7]. The proposed method STDAENN can provide a convenient hardware implementation of the sensing mechanism and a fast reconstruction with better reconstruction quality than the O-NL-SDA and BP.

**Table-3. Reconstruction Performance on
Different algorithms in the Labelme[20] dataset**

Images	O-NL-SDA[7] dB	BP[3] dB	STDAENN dB
Birds	26.62	22.75	26.75
Rabbit	27.24	22.63	28.51
Dog	21.55	16.97	26.47
Mean PSNR	26.34	22.07	27.24

5. CONCLUSION

The STDAENN method is proposed for a pair of relatively sparse ternarius(Latin word) projection matrix and a reconstruction method for compressed sensing of the image and that projection matrix could be used in hardware implementations very efficiently. An experimental outcome on the real data shows that the reconstruction performance based on PSNR for a projection matrix with 0.05 non-zero binary entries could be achieved and yield the best result compared with the O-NL-SDA and BP method. Sparse projection matrices with 0.001 non-zero entries realized with the help of STDAENN algorithm that gives an applicable performance as well. This algorithm is prolonged for implementation of MRI clinical system in this research.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests.

REFERENCES

- [1] D.L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory.* 52 (2006), 1289–1306.
- [2] Lu Gan, Block Compressed Sensing of Natural Images, in: 2007 15th International Conference on Digital Signal Processing, IEEE, Cardiff, UK, 2007: pp. 403–406.
- [3] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic Decomposition by Basis Pursuit, *SIAM Rev.* 43 (2001), 129–159.
- [4] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin, A Simple Proof of the Restricted Isometry Property for Random Matrices, *Constr. Approx.* 28 (2008), 253–263.
- [5] C. Dong, C.C. Loy, K. He, X. Tang, Image Super-Resolution Using Deep Convolutional Networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2016), 295–307.
- [6] P. Vincent, H. Larochelle, I. Lajoie, et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion, *J. Mach. Learn. Res.* 11 (2010), 3371–3408.
- [7] A. Mousavi, A.B. Patel, R.G. Baraniuk, A deep learning approach to structured signal recovery, in: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, Monticello, IL, 2015:

pp. 1336–1343.

- [8] A. Adler, D. Boubilil, M. Elad, M. Zibulevsky, A Deep Learning Approach to Block-based Compressed Sensing of Images, ArXiv:1606.01519 [Cs]. (2016).
- [9] M. Elad, Optimized Projections for Compressed Sensing, *IEEE Trans. Signal Process.* 55 (2007), 5695–5702.
- [10] E.V. Tsiligiani, L.P. Kondi, A.K. Katsaggelos, Construction of Incoherent Unit Norm Tight Frames With Application to Compressed Sensing, *IEEE Trans. Inform. Theory.* 60 (2014), 2319–2330.
- [11] L. Applebaum, S.D. Howard, S. Searle, R. Calderbank, Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery, *Appl. Comput. Harmon. Anal.* 26 (2009), 283–290.
- [12] J. Haupt, W.U. Bajwa, G. Raz, R. Nowak, Toeplitz Compressed Sensing Matrices With Applications to Sparse Channel Estimation, *IEEE Trans. Inform. Theory.* 56 (2010), 5862–5875.
- [13] M. Courbariaux, Y. Bengio, J.P. David, Binaryconnect: training deep neural networks with binary weights during propagations. In: *Advances in Neural Information Processing Systems*, pp. 3105–3113 (2015).
- [14] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1, ArXiv:1602.02830 [Cs]. (2016).
- [15] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, 2016: pp. 525–542.
- [16] S. Han, J. Pool, J. Tran, W. Dally, Learning both weights and connections for efficient neural network. In: *Advances in Neural Information Processing Systems*, pp. 1135–1143 (2015).
- [17] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323, (2011).
- [18] D.M. Christilin, A. Brighty, M.S. Mary, Compressed Sensing Reconstruction based on Adaptive Scale Parameter using Texture Feature. *Int. J. Innov. Technol. Explor. Eng.* 8(12) (2019), 5455–5461.
- [19] O. Russakovsky, J. Deng, H. Su, et al. ImageNet Large Scale Visual Recognition Challenge, *Int. J. Comput. Vis.* 115 (2015), 211–252.
- [20] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, LabelMe: A Database and Web-Based Tool for Image Annotation, *Int. J. Comput. Vis.* 77 (2008), 157–173.

- [21] Y. Burda, R. Grosse, R. Salakhutdinov, Importance Weighted Autoencoders, ArXiv:1509.00519 [Cs, Stat]. (2016).
- [22] S. Li, G. Ge, Deterministic Construction of Sparse Sensing Matrices via Finite Geometry, IEEE Trans. Signal Process. 62 (2014), 2850–2859.
- [23] A. Amini, F. Marvasti, Deterministic Construction of Binary, Bipolar, and Ternary Compressed Sensing Matrices, IEEE Trans. Inform. Theory. 57 (2011), 2360–2370.